

Programmieren mit Jython – Teil 3: Variablen

Lernziele

- Verwendung von Variablen in Programmen und zur Anpassung von Befehlen.
- Versetzen der Turtle an eine bestimmten Position im Koordinatensystem.

Theorie

Bis jetzt waren die Seitenlängen im Programmcode fest eingebaut. Wir können aber auch Programme schreiben, bei denen die Seitenlängen nach dem Programmstart verändert werden. Dazu benutzt man Variablen. Variablen haben einen Namen und einen Wert. Den Namen der Variable darf man frei wählen. Nicht erlaubt sind allerdings Befehlswörter und Namen mit Sonderzeichen. Einer Variablen kann mit dem Gleichheitszeichen ein Wert zugewiesen werden (z.B. $s = 60$). Ihr Wert kann jederzeit mit einer neuen Zuweisung verändert werden (z.B. $s = s + 10$).

Aufgaben

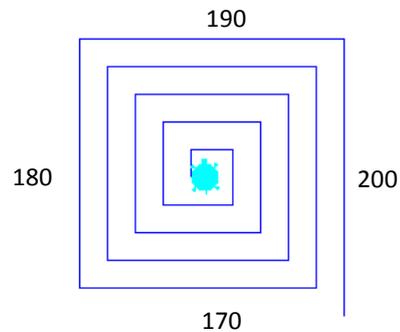
3.1) Schreibe folgendes Programm und führe es aus:

```
from gturtle import *
makeTurtle()

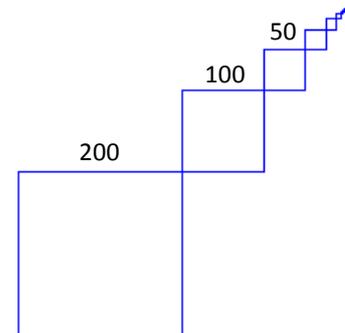
seite = 20
repeat 3:
    repeat 4:
        forward(seite)
        left(90)
        seite = seite + 20
```

Was macht dieses Programm?
Welches ist die Variable und welche Werte nimmt sie im Laufe des Programmes an?

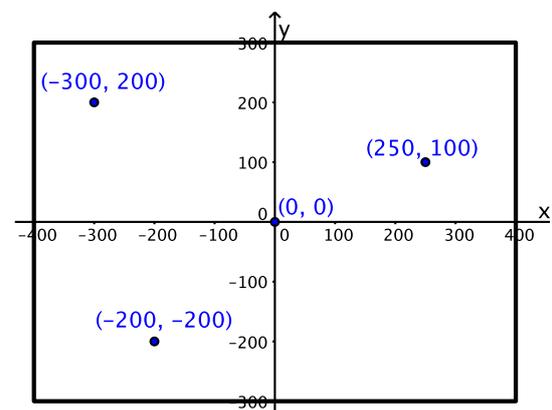
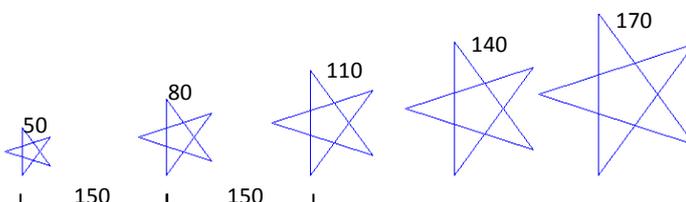
3.2) Zeichne folgende Figur möglichst effizient, indem Du bei der Programmierung Variablen verwendest:



3.3) Zeichne nebenstehende Figur. Führe als erstes den Befehl `setPos(-200, -200)` aus. Damit wird die Turtle zu Beginn auf eine Position links unterhalb der Mitte des Turtle-Fensters gesetzt. Das Zahlenpaar $(-200, -200)$ entspricht den Koordinaten eines Punktes im Koordinatensystem des Turtle-Fensters (siehe Bild unten rechts).



3.4) Variablen lassen sich nicht nur für die Seitenlänge einer Figur verwenden, sondern beispielsweise auch für die Platzierung der Turtle. Mit dem Befehl `setPos(x, y)` kann man die Turtle auf einen variablen Punkt im Koordinatensystem stellen, man muss vorgängig nur x und y auf die gewünschten Werte setzen. Zeichne mit Hilfe von `repeat`-Schleifen und `setPos(x, y)` folgende Sterne.



Das Koordinatensystem im Turtle-Fenster der Größe 800x600.

3.5) Verwende die Befehle `dot (p)` und `penUp ()` um Folgendes zu zeichnen:

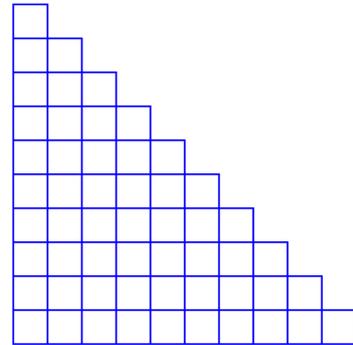


3.6) Zeichne das untenstehende Bild. Verändere dazu das Programm aus der letzten Aufgabe.



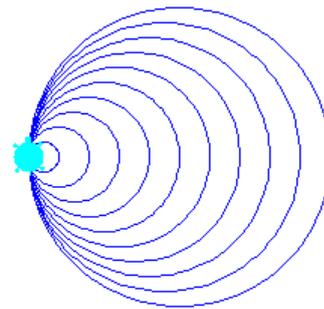
3.7) In Teil 2 des Unterrichts haben wir das Unterprogramm `QuadratZeile (Groesse, Anzahl)` kennengelernt.

Benutze dieses Unterprogramm und eine Variable, um mit einer Repeat-Schleife nebenstehende Figur zu zeichnen.

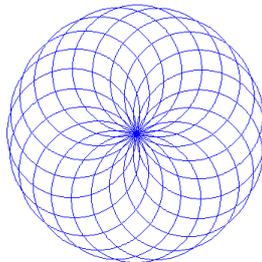


Zusatzaufgaben:

3.8) Zeichne die folgende Figur. Verwende hierfür den Befehl `rightArc (x, 360)`.



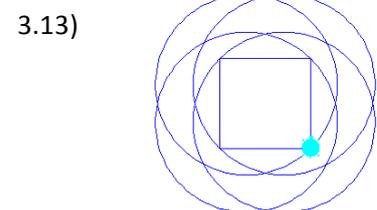
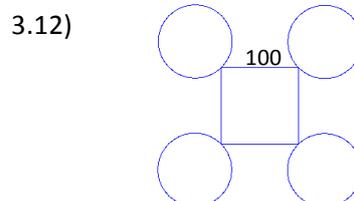
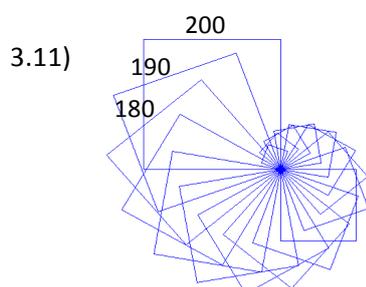
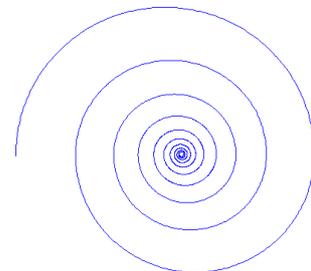
3.9) Zeichne folgende Figur:



20 Kreise mit Radius 100

3.10) Verändere Deinen Programmcode von 3.8) so, dass nur noch ein Halbkreis entsteht und zeichne damit folgende Figur aus Halbkreisen mit folgenden Werten:

- Start-Radius = 200
- Jeder folgende Radius misst 80% des vorhergehenden.
- Insgesamt 20 Halbkreise.



Hausaufgaben

Löse folgende Aufgaben mit **Papier und Bleistift** (ohne Computer):

H3.1) Was zeichnet folgendes Programm? Zeichne die Figur in den karierten Ausschnitt. Jedes Häuschen misst 10 Einheiten. Zu Beginn befindet sich die Turtle im eingezeichneten Punkt und schaut in Pfeilrichtung.

```
x=10
repeat 4:
  repeat 2:
    fd(x)
    rt(90)
  repeat 2:
    fd(x)
    lt(90)
x=x+10
```



H3.2) Betrachte folgendes Programm. Welchen Wert hat die Variable s am Schluss?

```
a=5
b=3
s=0
repeat a:
  repeat b:
    s=s+1
```

H3.3) Kunigunde wollte folgende Figur zeichnen und schrieb dazu das untenstehende Programm. Darin sind zwei Fehler enthalten. Finde diese.

```
laenge=100
right(90)
repeat 5:
  repeat 4:
    forward(x)
    left(120)
  right(120)
x-20=x
```

